CLAIMS

1. A system comprising:

a content host configured to receive multimedia contents for delivery to a content requester over a network, and to process the multimedia contents to provide an abstract content model that represents the multimedia contents; and

a decision engine communicatively linked with the content host and configured to receive the abstract content model and process the abstract content model to find an optimized sub-graph, and to select a delivery plan for the multimedia contents based at least in part on the optimized sub-graph.

2. The system of claim 1, wherein the content host is configured to process the multimedia contents to provide the abstract content model responsive to receiving a request for content from the content requester.

3. The system of claim 1, wherein the abstract content model is configured to hide physical details of the multimedia contents.

4. The system of claim 1 further comprising a resource model module configured to model network characteristics and to be used by the decision engine to select a delivery plan.

5.   The system of claim 1 further comprising a preference model module configured to model client capabilities and to be used by the decision engine to select a delivery plan.

6.   The system of claim 1 further comprising:

a resource model module configured to model network characteristics and to be used by the decision engine to select a delivery plan; and

a preference model module configured to model client capabilities and to be used by the decision engine to select a delivery plan.

7.   The system of claim 1, wherein the content host defines a set of application programming interfaces for retrieving extended properties of the abstract content model.

8.   The system of claim 1, wherein the decision engine is content independent.

9.   The system of claim 1 further comprising a content mapper module that is configured to process the abstract content model to provide multimedia content for delivery in accordance with the delivery plan.

10.   The system of claim 1, wherein the decision engine is configured to find the optimized sub-graph in a manner that maximizes one or more quality of service (QoS) values.

11. The system of claim 1, wherein the decision engine is configured to find the optimized sub-graph in a manner that maximizes one or more quality of service (QoS) values at least some of which being associated with resource constraints.

12. The system of claim 1, wherein the decision engine is configured to find the optimized sub-graph in a manner that maximizes one or more quality of service (QoS) values at least some of which being associated with client preference factors.

13. The system of claim 1, wherein the decision engine is configured to find the optimized sub-graph in a manner that maximizes one or more quality of service (QoS) values at least some of which being associated with resource constraints and others of which being associated with client preference factors.

14. The system of claim 1 further comprising a cache model configured to save at least portions of one or more selected delivery plans for reuse in other content request situations.

15. One or more computer-readable media embodying the system of claim 1.

16. One or more server computers embodying the system of claim 1.

17. A method comprising:

receiving a content request from a content requester;

retrieving the requested content from a content source;

processing the retrieved content to provide an abstract content model;

processing the abstract content model to select an optimal delivery plan the use of which will permit requested content to be delivered to the content requester; and

processing the abstract content model to provide deliverable content in accordance with the selected delivery plan.

18. The method of claim 17, wherein said processing of the retrieved content to provide the abstract content model comprises providing at least one directional graph.

19. The method of claim 18, wherein the directional graph features a top-down hierarchical structure.

20. The method of claim 18, wherein the directional graph comprises nodes that represent components of the content and edges that represent relationships between the nodes.

21. The method of claim 20, wherein the nodes are configured to have a node status that defines dynamic statuses of nodes during content delivery.

22. The method of claim 21, wherein at least one status comprises an inactive status where the node is not yet a deliverable object.

23. The method of claim 21, wherein at least one status comprises an activable status wherein an active condition of the node is satisfied but the node is not yet included in a delivery plan.

24. The method of claim 21, wherein at least one status comprises an activated status wherein the node has been chosen in a delivery plan.

25. The method of claim 21, wherein at least one status comprises a delivered status wherein the node has been delivered successfully to a content receiver.

26. The method of claim 21, wherein at least one status comprises a skipped status wherein the node is not delivered and will not be included in the delivery plan.

27. The method of claim 20, wherein there are multiple different types of edges.

28. The method of claim 27, wherein the multiple different types of edges comprise a dependency edge type that defines a logical dependency between nodes.

29. The method of claim 27, wherein the multiple different types of edges comprise a route edge type that defines an ordered or hierarchical dependency between nodes.

30. The method of claim 27, wherein the multiple different types of edges comprise a mixed edge type that defines a logical dependency between nodes and an ordered or hierarchical dependency between nodes.

31. The method of claim 27, wherein the multiple different types of edges comprise one or more of the following:

a dependency edge type that defines a logical dependency between nodes,

a route edge type that defines an ordered or hierarchical dependency between nodes, and

a mixed edge type that defines a logical dependency between nodes and an ordered or hierarchical dependency between nodes.

32. The method of claim 20, wherein the directional graph comprises a directional graph $G=(N, E)$ that satisfies the following condition, where N and E stand for "node" and "edge" sets of G:

- The node set N can be divided into several subsets as N_i,

$i=1..m$ where $N = \bigcup_{i=1}^{m} N_i$ and $N_i \bigcap N_j = \Phi$,

$\forall i, j = 1..m, i \neq j$.

- $\forall a = (n_s, n_e) \in E$, there exists a pair of $1 \leq i < j \leq m$ such that $n_s \in N_i$ and $n_e \in N_j$.

- $\forall a = (n_s n_e) \in E, s \neq e.$

33. A server computer programmed to implement the method of claim 17.

34. A method comprising:

processing content that can be the subject of a content request from a content requester to provide an abstract content model;

processing the abstract content model to select at least one delivery plan the use of which will permit requested content to be delivered to a content requester; and

processing the abstract content model to provide deliverable content in accordance with the selected delivery plan.

35. The method of claim 34, wherein said processing of the retrieved content to provide the abstract content model comprises providing at least one directional graph.

36. The method of claim 35, wherein the directional graph features a top-down hierarchical structure.

37. The method of claim 35, wherein the directional graph comprises nodes that represent components of the content and edges that represent relationships between the nodes.

38. A method comprising:

receiving a content request from a content requester;

retrieving the requested content from a content source;

processing the retrieved content to provide an abstract content model comprising a directional graph featuring a top-down hierarchical structure having nodes that represent components of the content and edges that represent relationships between the nodes, the nodes being configured to have a node status that defines dynamic statuses of nodes during content delivery, the node statuses being selected from a group of statuses comprising: (1) inactive status where the node is not yet a deliverable object, (2) activable status wherein an active condition of the node is satisfied but the node is not yet included in a delivery plan, (3) activated status wherein the node has been chosen in a delivery plan, (4) delivered status wherein the node has been delivered successfully to a content receiver, and (5) skipped status wherein the node is not delivered and will not be included in the delivery plan; and wherein there are multiple different types of edges selected from a group of types comprising: (1) a dependency edge type that defines a logical dependency between nodes, (2) a route edge type that defines an ordered or hierarchical dependency between nodes, and (3) a mixed edge type that defines a logical dependency between nodes and an ordered or hierarchical dependency between nodes;

processing the abstract content model to select an optimal delivery plan the use of which will permit requested content to be delivered to the content requester; and

processing the abstract content model to provide deliverable content in accordance with the selected delivery plan.

39. The method of claim 38, wherein the abstract content model comprises an ignition edge that is defined as a dependency edge from a node that is activated, delivered or skipped.

40. The method of claim 39, wherein nodes have active conditions that define how a node becomes activable.

41. The method of claim 39, wherein nodes have active conditions that define how a node becomes activable, one of the conditions comprising an automatic condition wherein the node is automatically activable.

42. The method of claim 39, wherein nodes have active conditions that define how a node becomes activable, one of the conditions comprising an OR condition wherein the node is activable if at least one of its input edges is an ignition edge.

43. The method of claim 39, wherein nodes have active conditions that define how a node becomes activable, one of the conditions comprising an AND condition wherein the node is activable only if all its input edges are ignition edges.

44. The method of claim 39, wherein nodes can have input conditions that define when an activable node can be activated.

45. The method of claim 39, wherein nodes can have output behaviors that define how nodes on ends of outgoing route edges can be branched in accordance with a branching operation, where a branched node constitutes a node that has been changed from an activable node to an activated node.

46. The method of claim 45, wherein there are multiple different branching operations.

47. The method of claim 46, wherein one of the multiple different branching operations comprises a complementary branching operation wherein all activable nodes can be branched.

48. The method of claim 46, wherein one of the multiple different branching operations comprises an exclusive branching operation wherein only one node from all activable nodes can be branched.

49. The method of claim 46, wherein one of the multiple different branching operations comprises a tight branching operation wherein all activable nodes must be branched.

50. The method of claim 38, wherein nodes can have a value associated with a Quality of Service (QoS) factor which defines an increment to content quality when this node has been delivered.

51. The method of claim 38, wherein nodes can have a resource factor that defines an amount of resources needed to deliver the node.

52. A data structure comprising:

an abstract content model that represents content that can be requested from a content requester, the abstract model comprising a directional graph that features a top-down hierarchical structure having nodes that represent components of the content and edges that represent relationships between the nodes, the nodes being configured to have a node status that defines dynamic statuses of the nodes during content delivery;

the abstract content model being configured for processing so that an optimal delivery plan for delivering the content can be selected.

53. The data structure of claim 52, wherein at least one status comprises an inactive status where the node is not yet a deliverable object.

54. The data structure of claim 52, wherein at least one status comprises an activable status wherein an active condition of the node is satisfied but the node is not yet included in a delivery plan.

55. The data structure of claim 52, wherein at least one status comprises an activated status wherein the node has been chosen in a delivery plan.

56. The data structure of claim 52, wherein at least one status comprises a delivered status wherein the node has been delivered successfully to a content receiver.

57. The data structure of claim 52, wherein at least one status comprises a skipped status wherein the node is not delivered and will not be included in the delivery plan.

58. The data structure of claim 52, wherein there are multiple different types of edges.

59. The data structure of claim 58, wherein the multiple different types of edges comprise a dependency edge type that defines a logical dependency between nodes.

60. The data structure of claim 58, wherein the multiple different types of edges comprise a route edge type that defines an ordered or hierarchical dependency between nodes.

61. The data structure of claim 58, wherein the multiple different types of edges comprise a mixed edge type that defines a logical dependency between nodes and an ordered or hierarchical dependency between nodes.

62.   The data structure of claim 58, wherein the multiple different types of edges comprise one or more of the following:

a dependency edge type that defines a logical dependency between nodes,

a route edge type that defines an ordered or hierarchical dependency between nodes, and

a mixed edge type that defines a logical dependency between nodes and an ordered or hierarchical dependency between nodes.